Compact and Malicious ***Private Set Intersection*** *for Small Sets*

Mike Rosulek, Oregon State University
Ni Trieu, Arizona State University

appeared at ACM CCS 2021

what is private set intersection (PSI)?

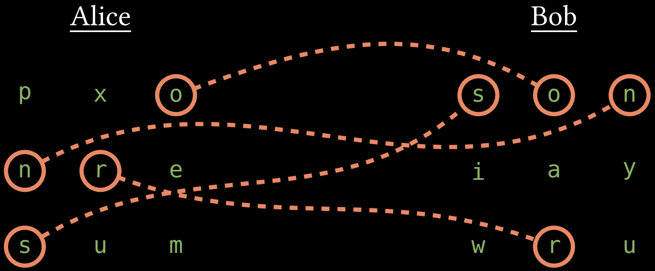
Alice

```
p   x   o

n   r   e

s   u   m
```

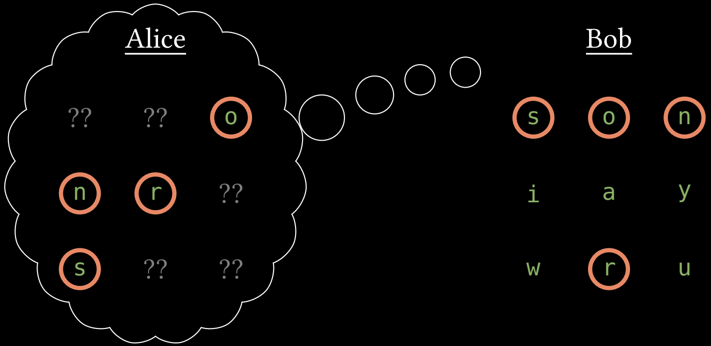
Bob

```
s   o   n

i   a   y

w   r   u
```

what is private set intersection (PSI)?



what is private set intersection (PSI)?



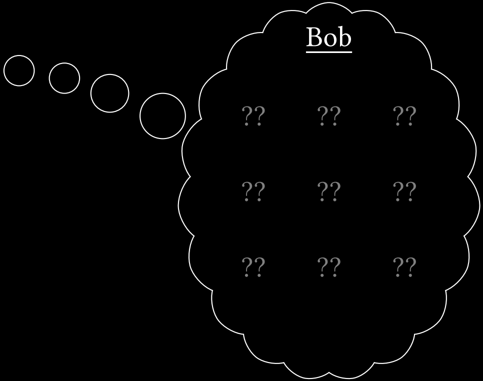
what is private set intersection (PSI)?

Alice

p x o

n r e

s u m



Bob

?? ?? ??

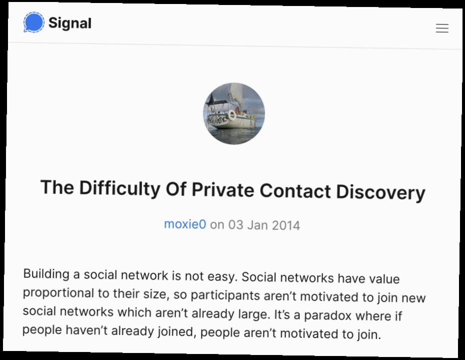
?? ?? ??

?? ?? ??

(one-sided output)

motivating PSI

motivating PSI



Signal

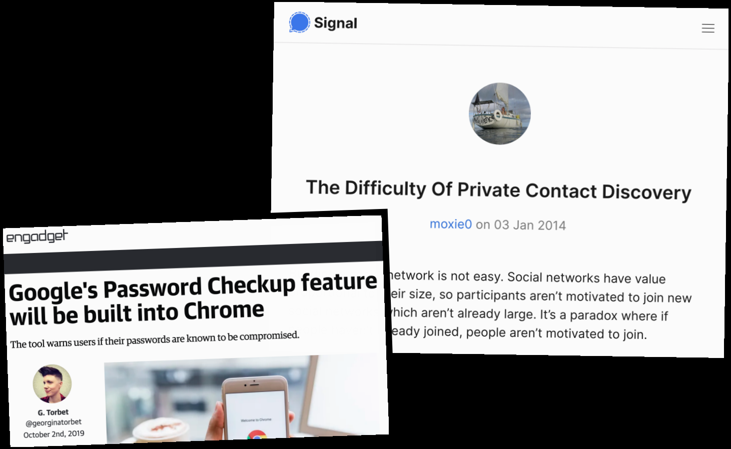
The Difficulty Of Private Contact Discovery

moxie0 on 03 Jan 2014

Building a social network is not easy. Social networks have value proportional to their size, so participants aren't motivated to join new social networks which aren't already large. It's a paradox where if people haven't already joined, people aren't motivated to join.

{my phone contacts} ∩ {users of your service}

motivating PSI



Signal

The Difficulty Of Private Contact Discovery

moxie0 on 03 Jan 2014

etwork is not easy. Social networks have value
eir size, so participants aren't motivated to join new
hich aren't already large. It's a paradox where if
eady joined, people aren't motivated to join.

engadget

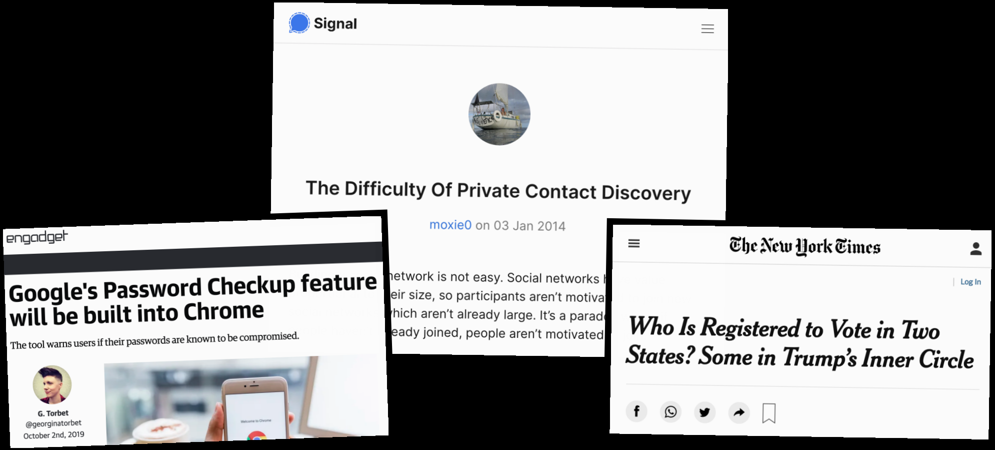
Google's Password Checkup feature will be built into Chrome

The tool warns users if their passwords are known to be compromised.

G. Torbet
@georginatorbet
October 2nd, 2019

{my passwords} ∩ {passwords found in breaches}

motivating PSI



Signal

The Difficulty Of Private Contact Discovery

moxie0 on 03 Jan 2014

etwork is not easy. Social networks h
eir size, so participants aren't motiva
hich aren't already large. It's a parad
eady joined, people aren't motivated

engadget

Google's Password Checkup feature will be built into Chrome

The tool warns users if their passwords are known to be compromised.

G. Torbet
@georginatorbet
October 2nd, 2019

The New York Times

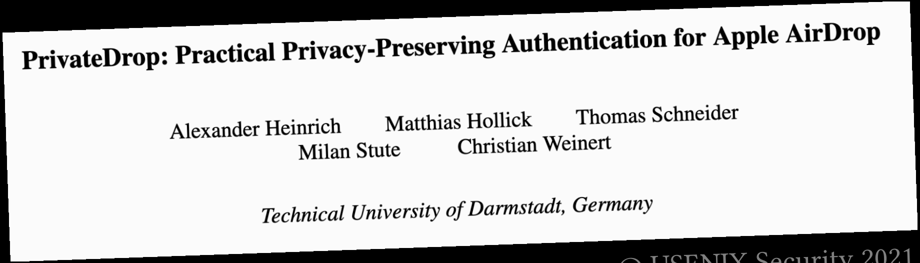
Log In

Who Is Registered to Vote in Two States? Some in Trump's Inner Circle

$$\{\text{voters registered in OR}\} \cap \{\text{voters registered in NY}\}$$

motivating PSI for small sets

motivating PSI for small sets



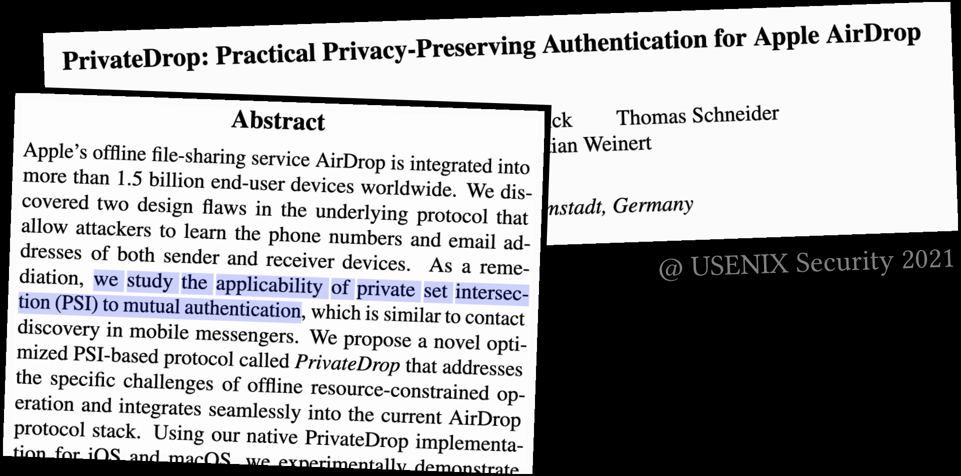
PrivateDrop: Practical Privacy-Preserving Authentication for Apple AirDrop

Alexander Heinrich Matthias Hollick Thomas Schneider
Milan Stute Christian Weinert

Technical University of Darmstadt, Germany

@ USENIX Security 2021

motivating PSI for small sets



PrivateDrop: Practical Privacy-Preserving Authentication for Apple AirDrop

ck Thomas Schneider
ian Weinert

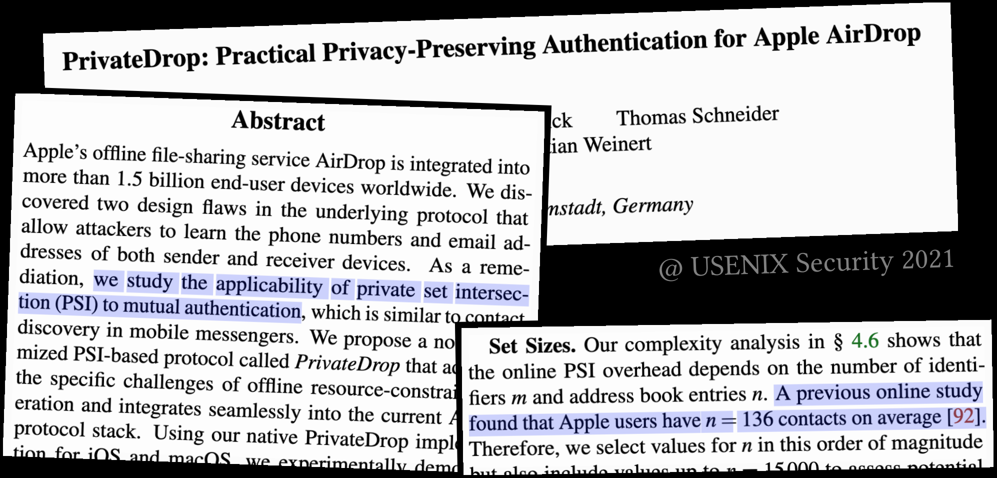
nstadt, Germany

Abstract

Apple's offline file-sharing service AirDrop is integrated into more than 1.5 billion end-user devices worldwide. We discovered two design flaws in the underlying protocol that allow attackers to learn the phone numbers and email addresses of both sender and receiver devices. As a remediation, we study the applicability of private set intersection (PSI) to mutual authentication, which is similar to contact discovery in mobile messengers. We propose a novel optimized PSI-based protocol called *PrivateDrop* that addresses the specific challenges of offline resource-constrained operation and integrates seamlessly into the current AirDrop protocol stack. Using our native PrivateDrop implementation for iOS and macOS, we experimentally demonstrate

@ USENIX Security 2021

motivating PSI for small sets



PrivateDrop: Practical Privacy-Preserving Authentication for Apple AirDrop

ck Thomas Schneider
ian Weinert

nstadt, Germany

Abstract

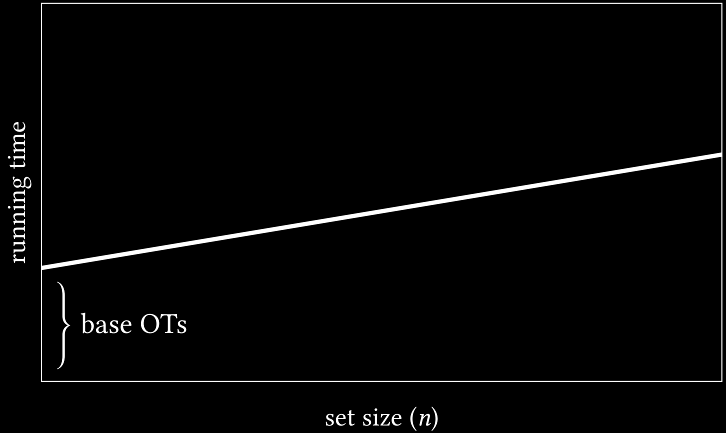
Apple's offline file-sharing service AirDrop is integrated into more than 1.5 billion end-user devices worldwide. We discovered two design flaws in the underlying protocol that allow attackers to learn the phone numbers and email addresses of both sender and receiver devices. As a remediation, we study the applicability of private set intersection (PSI) to mutual authentication, which is similar to contact discovery in mobile messengers. We propose a no mized PSI-based protocol called *PrivateDrop* that ad the specific challenges of offline resource-constrai eration and integrates seamlessly into the current A protocol stack. Using our native PrivateDrop imple tion for iOS and macOS, we experimentally demo

Set Sizes. Our complexity analysis in § 4.6 shows that the online PSI overhead depends on the number of identifiers m and address book entries n. A previous online study found that Apple users have $n = 136$ contacts on average [92]. Therefore, we select values for n in this order of magnitude but also include values up to $n = 15\,000$ to assess potential

@ USENIX Security 2021

[92] = Stute, Narain, Mariotto, Heinrich, Kreitschmann, Noubir, Hollick @ *USENIX* 2019

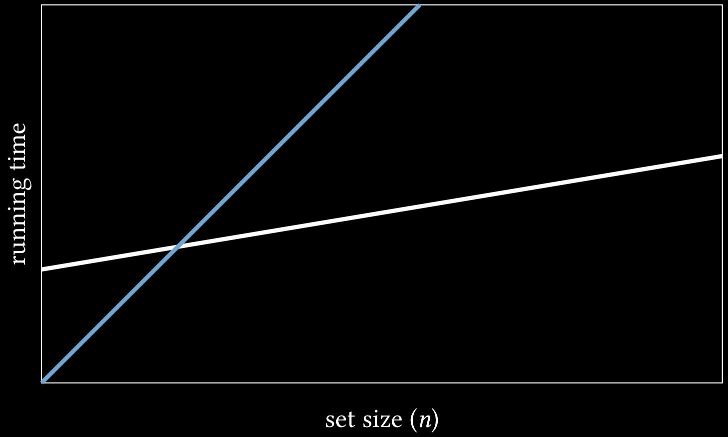
PSI techniques for small sets



OT-based PSI:

- 128 base OTs
- $O(n)$ symm-key ops

PSI techniques for small sets



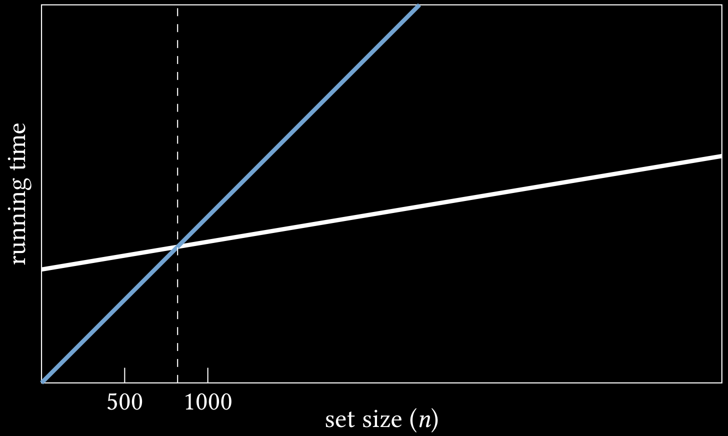
OT-based PSI:

- 128 base OTs
- $O(n)$ symm-key ops

KA-based PSI:

- $O(n)$ pub-key ops

PSI techniques for small sets



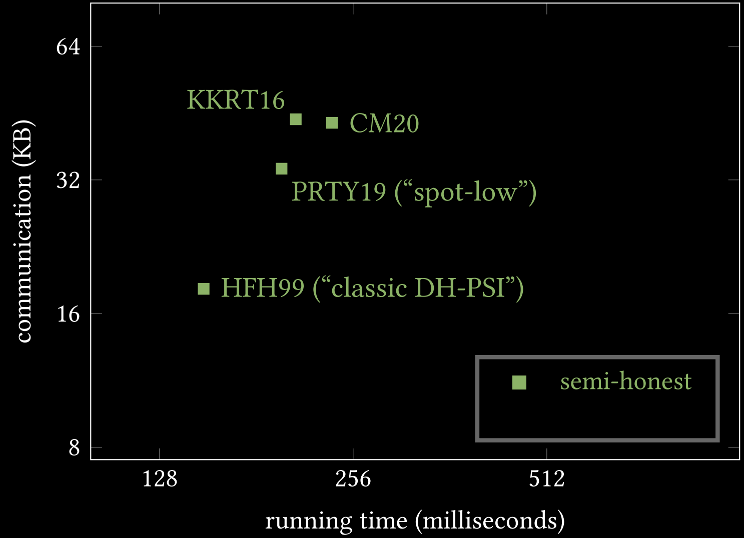
OT-based PSI:

- 128 base OTs
- $O(n)$ symm-key ops

KA-based PSI:

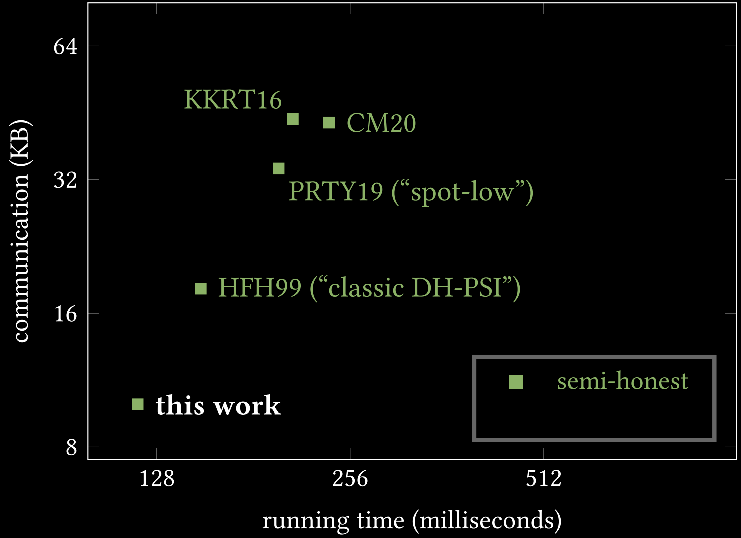
- $O(n)$ pub-key ops

PSI cost: 256 items per party:



our **semi-honest** PSI:

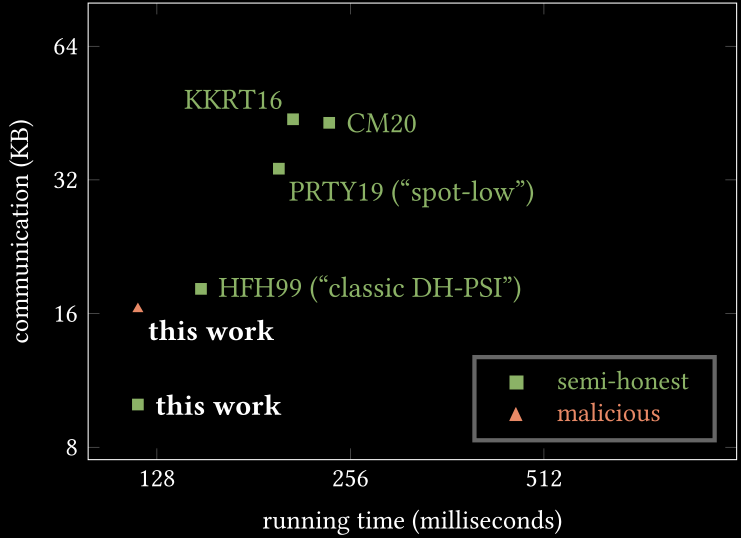
PSI cost: 256 items per party:



our **semi-honest** PSI:

- 45% ↓ communication
- 20% ↓ runtime

PSI cost: 256 items per party:



our **semi-honest** PSI:

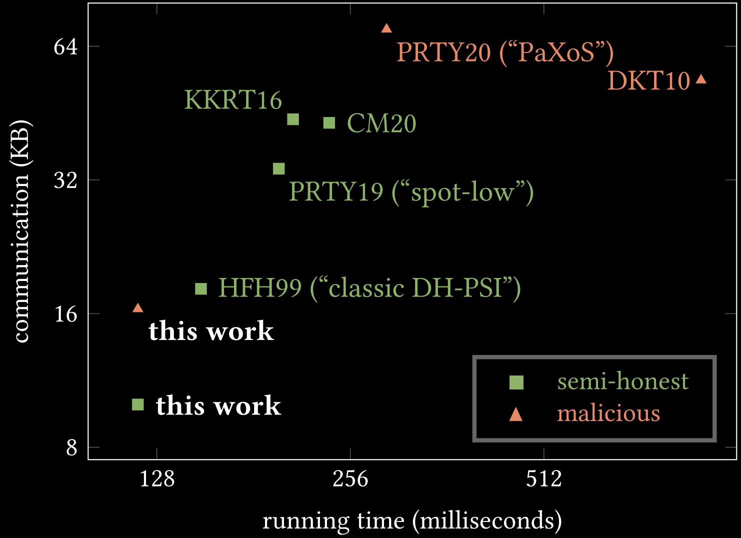
- 45% ↓ communication
- 20% ↓ runtime

our **malicious** PSI:

- 10% ↓ communication
- 20% ↓ runtime

vs. best *semi-honest* PSI!

PSI cost: 256 items per party:



our **semi-honest** PSI:

- 45% ↓ communication
- 20% ↓ runtime

our **malicious** PSI:

- 10% ↓ communication
- 20% ↓ runtime

vs. best *semi-honest* PSI!

- 75% ↓ communication
- 55% ↓ runtime

vs. best malicious PSI

✓ 1 *what is PSI?*

✓ 2 *summary of our results*

3 *review "classic DH-PSI"*

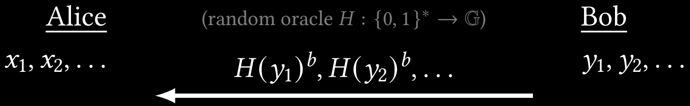
4 *our new protocol ideas*

5 *some fine print*

Alice (random oracle $H : \{0,1\}^* \to \mathbb{G}$) Bob

$x_1, x_2, \ldots$ $y_1, y_2, \ldots$

[HubermanFranklinHogg99]



Alice

(random oracle $H : \{0,1\}^* \to \mathbb{G}$)

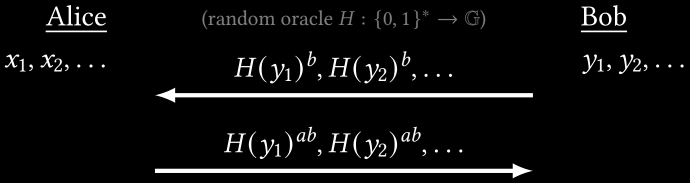
Bob

$x_1, x_2, \ldots$

$H(y_1)^b, H(y_2)^b, \ldots$

$y_1, y_2, \ldots$

[HubermanFranklinHogg99]



Alice (random oracle $H : \{0,1\}^* \to \mathbb{G}$) Bob

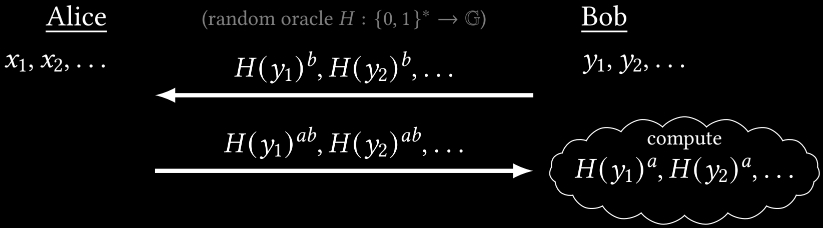
$x_1, x_2, \dots$

$y_1, y_2, \dots$

$$\xleftarrow{H(y_1)^b, H(y_2)^b, \dots}$$

$$\xrightarrow{H(y_1)^{ab}, H(y_2)^{ab}, \dots}$$

[HubermanFranklinHogg99]



Alice

(random oracle $H : \{0,1\}^* \rightarrow \mathbb{G}$)

Bob

$x_1, x_2, \ldots$

$y_1, y_2, \ldots$

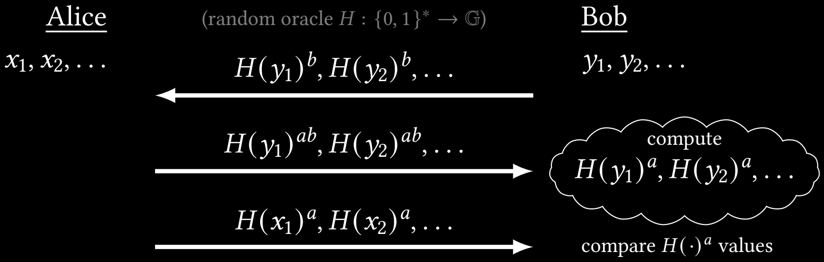
$H(y_1)^b, H(y_2)^b, \ldots$

compute

$H(y_1)^a, H(y_2)^a, \ldots$

$H(y_1)^{ab}, H(y_2)^{ab}, \ldots$

[HubermanFranklinHogg99]



Alice

(random oracle $H : \{0,1\}^* \to \mathbb{G}$)

Bob

$x_1, x_2, \ldots$

$y_1, y_2, \ldots$

$H(y_1)^b, H(y_2)^b, \ldots$

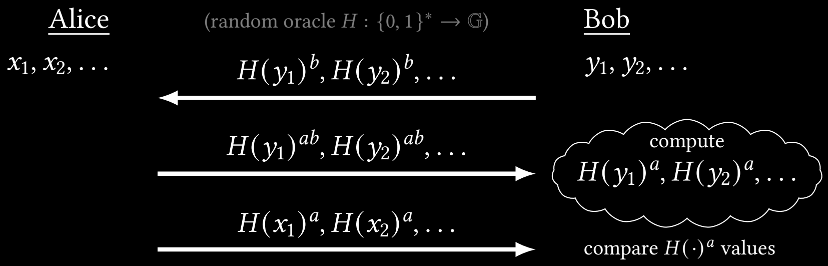
$H(y_1)^{ab}, H(y_2)^{ab}, \ldots$

compute $H(y_1)^a, H(y_2)^a, \ldots$

$H(x_1)^a, H(x_2)^a, \ldots$

compare $H(\cdot)^a$ values

[HubermanFranklinHogg99]



Alice

(random oracle $H : \{0,1\}^* \to \mathbb{G}$)

Bob

$x_1, x_2, \dots$

$y_1, y_2, \dots$

$H(y_1)^b, H(y_2)^b, \dots$

$H(y_1)^{ab}, H(y_2)^{ab}, \dots$

compute $H(y_1)^a, H(y_2)^a, \dots$

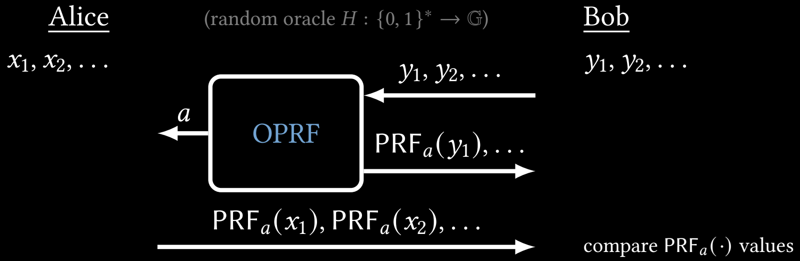
$H(x_1)^a, H(x_2)^a, \dots$

compare $H(\cdot)^a$ values

Semi-honest security:

- $x \mapsto H(x)^a$ is a PRF (DDH assumption + random oracle)
- first two messages are an oblivious PRF protocol

[HubermanFranklinHogg99]



Alice

(random oracle $H : \{0,1\}^* \to \mathbb{G}$)

Bob

$x_1, x_2, \ldots$

$y_1, y_2, \ldots$

$y_1, y_2, \ldots$

a

OPRF

$\mathsf{PRF}_a(y_1), \ldots$

$\mathsf{PRF}_a(x_1), \mathsf{PRF}_a(x_2), \ldots$

compare $\mathsf{PRF}_a(\cdot)$ values

Semi-honest security:

- $x \mapsto H(x)^a$ is a PRF (DDH assumption + random oracle)
- first two messages are an oblivious PRF protocol
- standard OPRF→PSI paradigm [FreedmanIshaiPinkasReingold05]

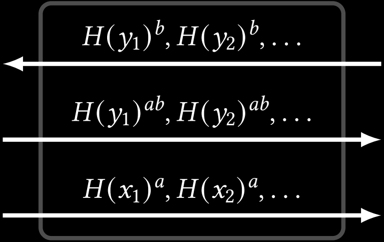
[HubermanFranklinHogg99]

Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

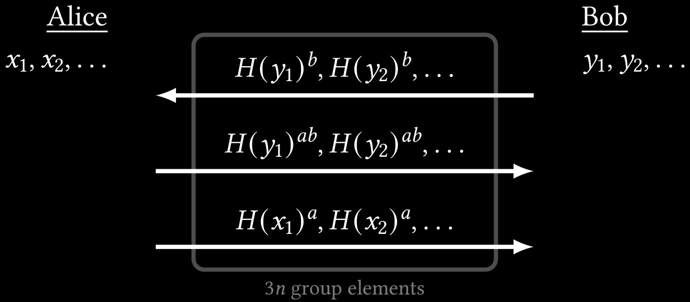


$H(y_1)^b, H(y_2)^b, \ldots$

$H(y_1)^{ab}, H(y_2)^{ab}, \ldots$

$H(x_1)^a, H(x_2)^a, \ldots$

$3n$ group elements



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

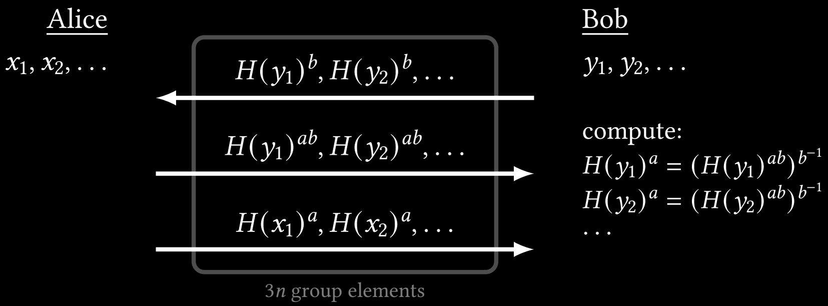
$H(y_1)^b, H(y_2)^b, \ldots$

$H(y_1)^{ab}, H(y_2)^{ab}, \ldots$

$H(x_1)^a, H(x_2)^a, \ldots$

$3n$ group elements

how could you possibly reduce communication?



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$H(y_1)^b, H(y_2)^b, \ldots$

$H(y_1)^{ab}, H(y_2)^{ab}, \ldots$

$H(x_1)^a, H(x_2)^a, \ldots$

$3n$ group elements

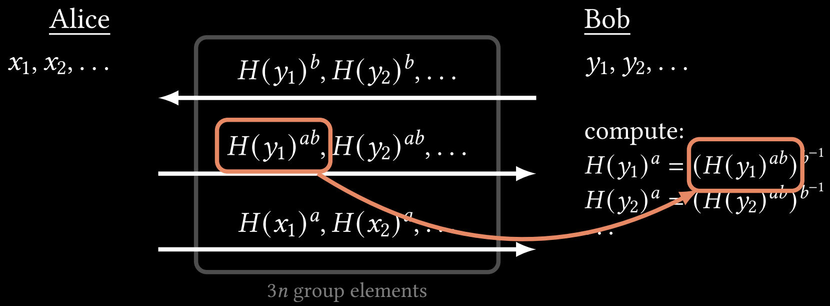
compute:

$H(y_1)^a = (H(y_1)^{ab})^{b^{-1}}$

$H(y_2)^a = (H(y_2)^{ab})^{b^{-1}}$

$\ldots$

how could you possibly reduce communication?



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$H(y_1)^b, H(y_2)^b, \ldots$

$\boxed{H(y_1)^{ab}}, H(y_2)^{ab}, \ldots$

$H(x_1)^a, H(x_2)^a, \ldots$

compute:

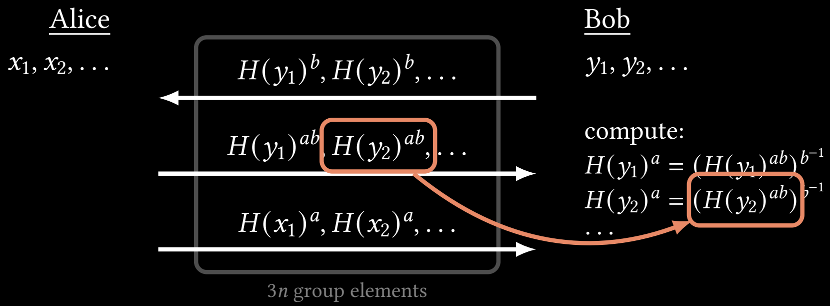
$H(y_1)^a = \boxed{(H(y_1)^{ab})}^{b^{-1}}$

$H(y_2)^a = (H(y_2)^{ab})^{b^{-1}}$

$\ldots$

$3n$ group elements

how could you possibly reduce communication?



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$H(y_1)^b, H(y_2)^b, \ldots$

$H(y_1)^{ab}, \boxed{H(y_2)^{ab}}, \ldots$

$H(x_1)^a, H(x_2)^a, \ldots$

compute:

$H(y_1)^a = (H(y_1)^{ab})^{b^{-1}}$

$H(y_2)^a = \boxed{(H(y_2)^{ab})}^{b^{-1}}$

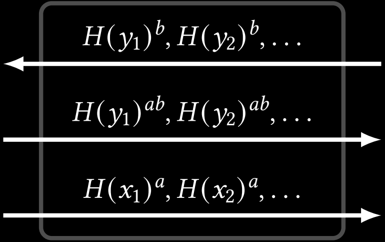
$\ldots$

$3n$ group elements

how could you possibly reduce communication?

Alice

$x_1, x_2, \ldots$



$H(y_1)^b, H(y_2)^b, \ldots$

$H(y_1)^{ab}, H(y_2)^{ab}, \ldots$

$H(x_1)^a, H(x_2)^a, \ldots$

3n group elements

Bob

$y_1, y_2, \ldots$

$H(y_1)^r = H(y_2)$

compute:

$H(y_1)^a = (H(y_1)^{ab})^{b^{-1}}$

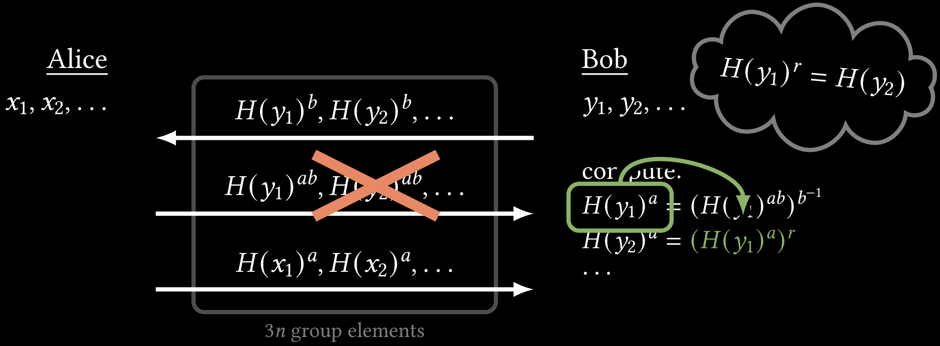
$H(y_2)^a = (H(y_2)^{ab})^{b^{-1}}$

$\ldots$

how could you possibly reduce communication?

replace random oracle with some "trapdoored" function

... where Bob knows dlog relationships between outputs



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$H(y_1)^r = H(y_2)$

$H(y_1)^b, H(y_2)^b, \ldots$

$\cancel{H(y_1)^{ab}, H(y_2)^{ab}, \ldots}$

$H(x_1)^a, H(x_2)^a, \ldots$

3n group elements

compute:

$\boxed{H(y_1)^a} = (H(y_1)^{ab})^{b^{-1}}$

$H(y_2)^a = (H(y_1)^a)^r$

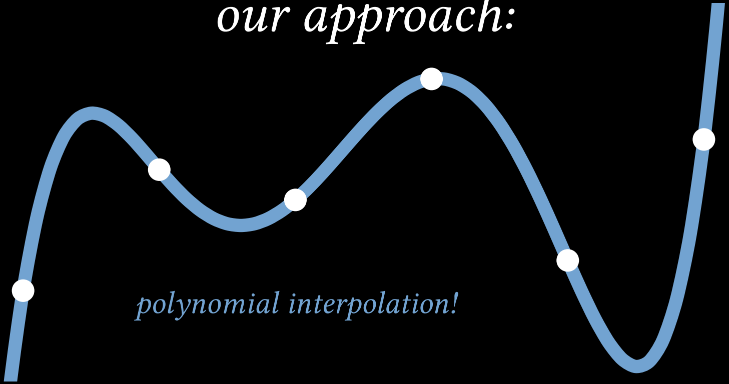
$\ldots$

how could you possibly reduce communication?

replace random oracle with some "trapdoored" function

... where Bob knows dlog relationships between outputs

our approach:



polynomial interpolation!

Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

Alice

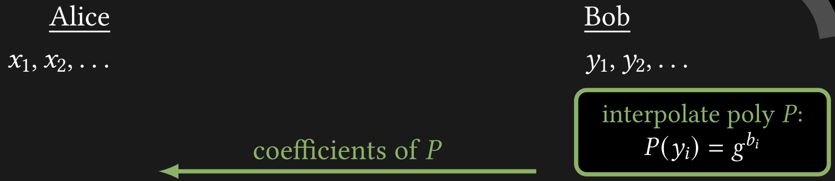
$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

interpolate poly P:

$P(y_i) = g^{b_i}$



Alice

$x_1, x_2, \ldots$

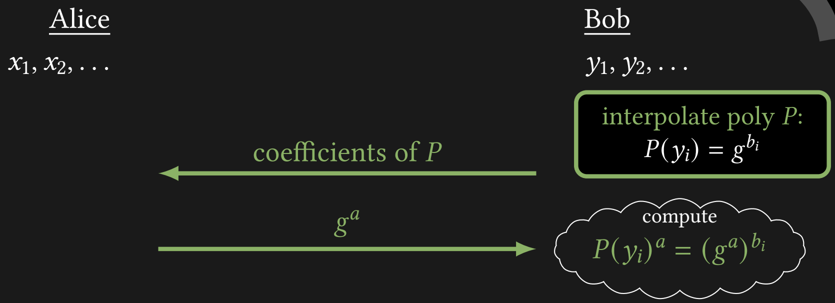
Bob

$y_1, y_2, \ldots$

interpolate poly P:

$P(y_i) = g^{b_i}$

coefficients of P



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

interpolate poly P:

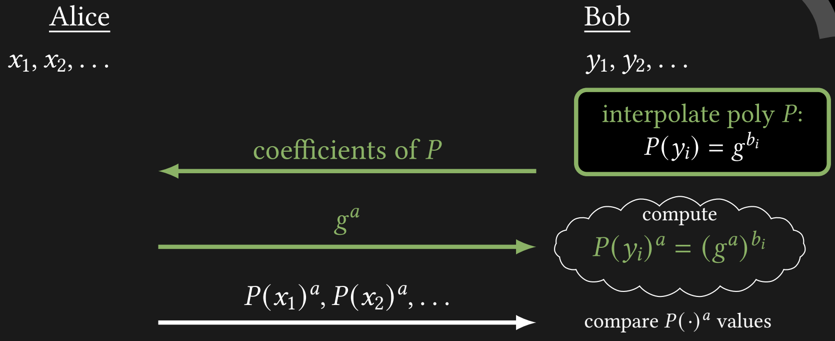
$P(y_i) = g^{b_i}$

coefficients of P

g^a

compute

$P(y_i)^a = (g^a)^{b_i}$



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

interpolate poly P:

$P(y_i) = g^{b_i}$

coefficients of P

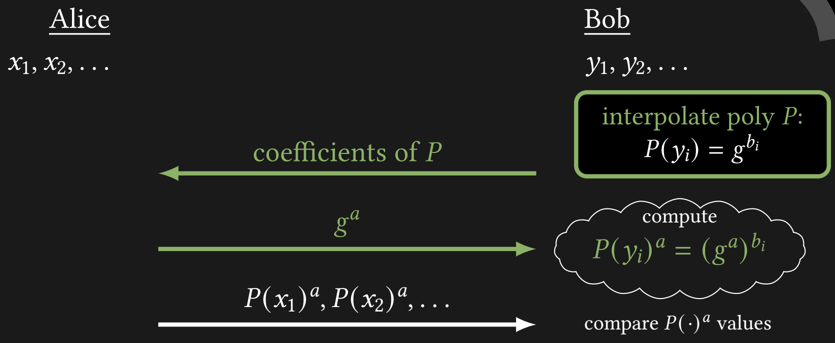
g^a

compute

$P(y_i)^a = (g^a)^{b_i}$

$P(x_1)^a, P(x_2)^a, \ldots$

compare $P(\cdot)^a$ values



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

interpolate poly P:
$P(y_i) = g^{b_i}$

coefficients of P

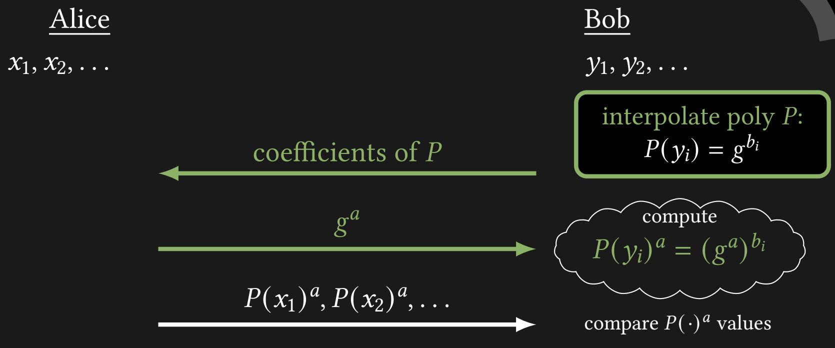
g^a

compute
$P(y_i)^a = (g^a)^{b_i}$

$P(x_1)^a, P(x_2)^a, \ldots$

compare $P(\cdot)^a$ values

correctness: Bob knows dlog of $P(y)$ for programmed points ✓



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

interpolate poly P:
$P(y_i) = g^{b_i}$

coefficients of P ←

g^a →

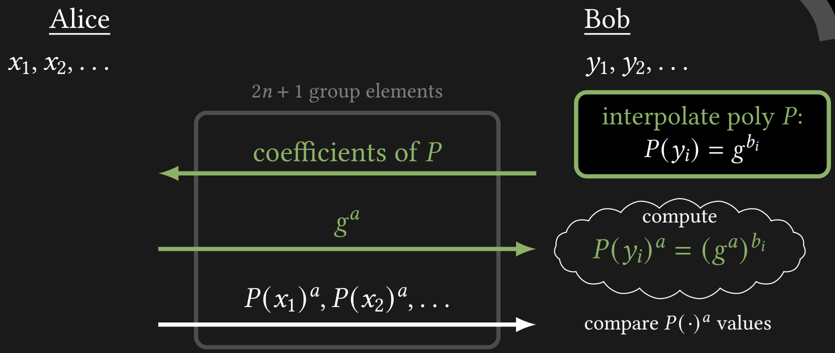
compute
$P(y_i)^a = (g^a)^{b_i}$

$P(x_1)^a, P(x_2)^a, \ldots$ →

compare $P(\cdot)^a$ values

correctness: Bob knows dlog of $P(y)$ for programmed points ✓

obliviousness: description of P doesn't leak choice of programmed points ✓



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$2n+1$ group elements

interpolate poly P: $P(y_i) = g^{b_i}$

coefficients of P

g^a

compute $P(y_i)^a = (g^a)^{b_i}$

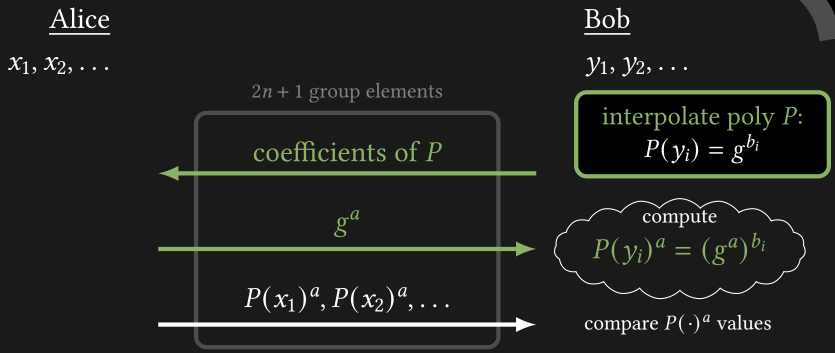
$P(x_1)^a, P(x_2)^a, \ldots$

compare $P(\cdot)^a$ values

correctness: Bob knows dlog of $P(y)$ for programmed points ✓

obliviousness: description of P doesn't leak choice of programmed points ✓

efficiency: $|\text{description of } P| = n$ group elements ✓



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$2n+1$ group elements

interpolate poly P:
$P(y_i) = g^{b_i}$

coefficients of P

g^a

$P(x_1)^a, P(x_2)^a, \ldots$

compute
$P(y_i)^a = (g^a)^{b_i}$

compare $P(\cdot)^a$ values

correctness:	Bob knows dlog of $P(y)$ for programmed points ✓
obliviousness:	description of P doesn't leak choice of programmed points ✓
efficiency:	$\vert$description of $P\vert = n$ group elements ✓
$P(\cdot)^a$ is PRF:	Bob **cannot know** dlog of any *other* $P(x)$??

interpolate so that:

$P(y_i) = g^{b_i}$

$? \; ? \; \Downarrow \; ? \; ?$

other $P(x)$ outputs have unknown dlog

interpolate so that:

$$P(y_i) = g^{b_i}$$

? ? $\Downarrow$? ?

other $P(x)$ outputs
have unknown dlog

Ideal permutation model: all parties have oracle access to random Π, Π^{-1}

interpolate so that:

$P(y_i) = g^{b_i}$

? ? $\Downarrow$? ?

other $P(x)$ outputs
have unknown dlog

interpolate so that:

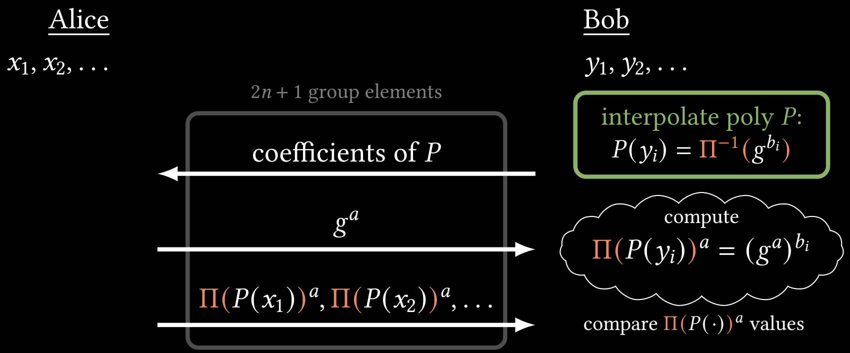
$P(y_i) = \Pi^{-1}(g^{b_i})$

$\Downarrow$ ✓

simulator can **program**
other $\Pi(P(x))$ outputs

Ideal permutation model: all parties have oracle access to random Π, Π^{-1}

our real protocol:



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$2n+1$ group elements

interpolate poly P:
$P(y_i) = \Pi^{-1}(g^{b_i})$

coefficients of P

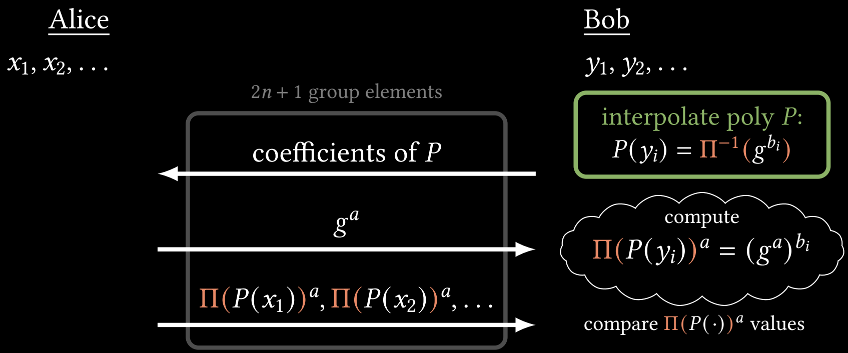
g^a

compute
$\Pi(P(y_i))^a = (g^a)^{b_i}$

$\Pi(P(x_1))^a, \Pi(P(x_2))^a, \ldots$

compare $\Pi(P(\cdot))^a$ values

our real protocol (fine print):



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

interpolate poly P:
$P(y_i) = \Pi^{-1}(g^{b_i})$

$2n + 1$ group elements

coefficients of P

g^a

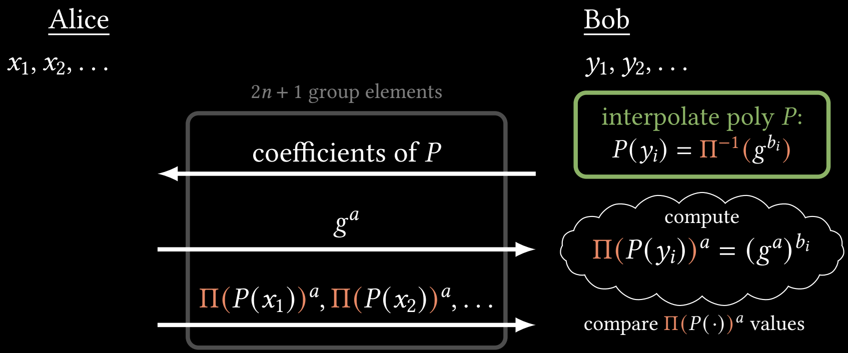
$\Pi(P(x_1))^a, \Pi(P(x_2))^a, \ldots$

compute
$\Pi(P(y_i))^a = (g^a)^{b_i}$

compare $\Pi(P(\cdot))^a$ values

semi-honest: Alice's group elements can be truncated

our real protocol (fine print):



Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$2n + 1$ group elements

interpolate poly P:

$P(y_i) = \Pi^{-1}(g^{b_i})$

coefficients of P

g^a

compute

$\Pi(P(y_i))^a = (g^a)^{b_i}$

$\Pi(P(x_1))^a, \Pi(P(x_2))^a, \ldots$

compare $\Pi(P(\cdot))^a$ values

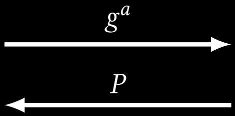
semi-honest: Alice’s group elements can be truncated

malicious: a few more strategic RO calls (to help simulator extract) *more fine print...*

$$\xrightarrow{g^a}$$

$$\xleftarrow{P}$$

$$P(y_i) = g^{b_i}$$



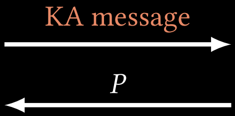
?

$$P(y_i) = g^{b_i} \quad ?$$

finite field

?

cyclic group

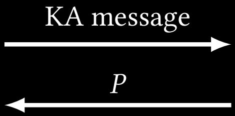


KA message

P

$$P(y_i) = \text{KA response}$$

- use generic key agreement in place of g^a, g^b



$$P(y_i) = \text{KA response}$$

- use generic key agreement in place of g^a, g^b
- KA protocol messages must be *pseudorandom bit strings*
- e.g., elliptic curve Diffie-Hellman with elligator encoding scheme [BernsteinHamburgKrasnovaLange13]

[ChoDachmanSoledJarecki16] PSI:

our protocol:

[ChoDachmanSoledJarecki16] PSI:

interpolate polynomial P so that:

$$P(y) = \boxed{\text{next message in private equality test protocol}}$$

our protocol:

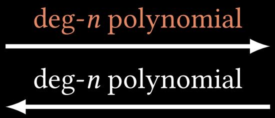
interpolate polynomial P so that:

$$P(y) = \boxed{\text{next message in key agreement protocol}}$$

[ChoDachmanSoledJarecki16] PSI:

interpolate polynomial P so that:

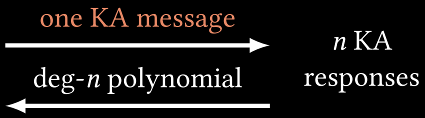
$$P(y) = \boxed{\begin{array}{l}\text{next message in private}\\ \text{equality test protocol}\end{array}}$$



our protocol:

interpolate polynomial P so that:

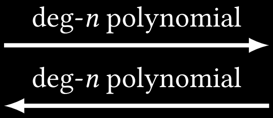
$$P(y) = \boxed{\begin{array}{l}\text{next message in key}\\ \text{agreement protocol}\end{array}}$$



[ChoDachmanSoledJarecki16] PSI:

interpolate polynomial P so that:

$$P(y) = \boxed{\begin{array}{l}\text{next message in private}\\ \text{equality test protocol}\end{array}}$$

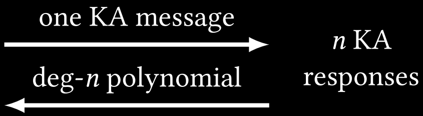


ideal cipher model

our protocol:

interpolate polynomial P so that:

$$P(y) = \boxed{\begin{array}{l}\text{next message in key}\\ \text{agreement protocol}\end{array}}$$

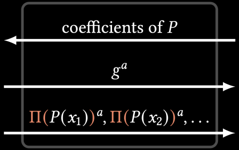


ideal permutation model

take-home message

new PSI protocol

$2n + 1$ group elements



coefficients of P

g^a

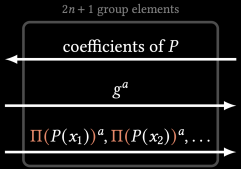
$\Pi(P(x_1))^a, \Pi(P(x_2))^a, \ldots$

ideal permutation + random oracle

full version @ `ia.cr/2021/1159`

take-home message

new PSI protocol



$2n+1$ group elements

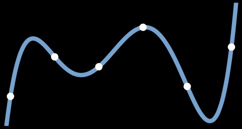
coefficients of P

g^a

$\Pi(P(x_1))^a, \Pi(P(x_2))^a, \ldots$

ideal permutation + random oracle

main protocol idea



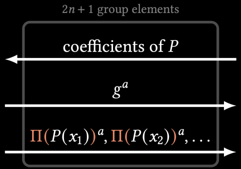
interpolate poly P so that

$P(y)$ = key agreement message

full version @ `ia.cr/2021/1159`

take-home message

new PSI protocol



$2n + 1$ group elements

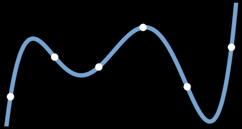
coefficients of P

g^a

$\Pi(P(x_1))^a, \Pi(P(x_2))^a, \ldots$

ideal permutation + random oracle

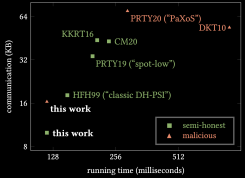
main protocol idea



interpolate poly P so that

$P(y)$ = key agreement message

performance on small (< 1000) sets

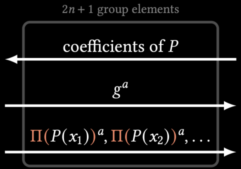


best by good margin

full version @ ia.cr/2021/1159

take-home message

new PSI protocol



$2n + 1$ group elements

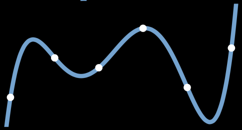
coefficients of P

g^a

$\Pi(P(x_1))^a, \Pi(P(x_2))^a, \ldots$

ideal permutation + random oracle

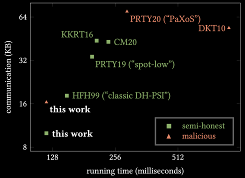
main protocol idea



interpolate poly P so that

$P(y)$ = key agreement message

performance on small (< 1000) sets



best by good margin

vs. 20-year old classic DH-PSI

faster ✓

less communication ✓

stronger security ✓

full version @ ia.cr/2021/1159